

Topologically Trivial Closed Walks in Directed Surface Graphs

Jeff Erickson 

University of Illinois at Urbana-Champaign, Urbana, IL, USA
jeffe@illinois.edu

Yipu Wang

University of Illinois at Urbana-Champaign, Urbana, IL, USA
ywang298@illinois.edu

Abstract

Let G be a directed graph with n vertices and m edges, embedded on a surface S , possibly with boundary, with first Betti number β . We consider the complexity of finding closed directed walks in G that are either contractible (trivial in homotopy) or bounding (trivial in integer homology) in S . Specifically, we describe algorithms to determine whether G contains a simple contractible cycle in $O(n + m)$ time, or a contractible closed walk in $O(n + m)$ time, or a bounding closed walk in $O(\beta(n + m))$ time. Our algorithms rely on subtle relationships between strong connectivity in G and in the dual graph G^* ; our contractible-closed-walk algorithm also relies on a seminal topological result of Hass and Scott. We also prove that detecting simple bounding cycles is NP-hard.

We also describe three polynomial-time algorithms to compute shortest contractible closed walks, depending on whether the fundamental group of the surface is free, abelian, or hyperbolic. A key step in our algorithm for hyperbolic surfaces is the construction of a context-free grammar with $O(g^2 L^2)$ non-terminals that generates all contractible closed walks of length at most L , and only contractible closed walks, in a system of quads of genus $g \geq 2$. Finally, we show that computing shortest simple contractible cycles, shortest simple bounding cycles, and shortest bounding closed walks are all NP-hard.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases computational topology, surface-embedded graphs, homotopy, homology, strong connectivity, hyperbolic geometry, medial axes, context-free grammars

Digital Object Identifier 10.4230/LIPIcs.SoCG.2019.34

Category Regular Paper

Related Version A full version of this paper is available at [arXiv:1812.01564](https://arxiv.org/abs/1812.01564) [30].

Funding Research on this paper was partially supported by NSF grant CCF-1408763.

Acknowledgements The first author would like to thank Tillmann Miltzow for asking an annoying question that led to this work, and to apologize for still being unable to answer it.

1 Introduction

A key step in several algorithms for surface-embedded graphs is finding a shortest closed walk and/or simple cycle in the input graph with some interesting topological property. There is a large body of work on finding short interesting walks and cycles in undirected surface graphs, starting with Thomassen's seminal *3-path condition* [49, 55]. For example, efficient algorithms are known for computing shortest non-contractible and non-separating cycles [7, 12, 27, 28, 46], shortest contractible closed walks [10], simple cycles that are shortest in their own homotopy class [11], and shortest closed walks in a *given* homotopy [18] or homology class [29], and for detecting simple cycles that are either contractible, non-contractible, or non-separating [9]. On the other hand, several related problems are known to be NP-hard, including computing



© Jeff Erickson and Yipu Wang;

licensed under Creative Commons License CC-BY

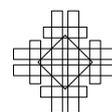
35th International Symposium on Computational Geometry (SoCG 2019).

Editors: Gill Barequet and Yusu Wang; Article No. 34; pp. 34:1–34:17

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



shortest splitting closed walks [13], computing shortest separating cycles [6], computing shortest closed walks in a given homology class [14], and deciding whether a surface graph contains a simple separating or splitting cycle [9].

Directed surface graphs are much less understood, in part because they do not share convenient properties of undirected graphs, such as Thomassen’s 3-path condition [49, 55], or the assumption that if shortest paths are unique, then two shortest paths cross at most once [26]. The first progress in this direction was a pair of algorithms by Cabello, Colin de Verdière, and Lazarus [8], which compute shortest non-contractible and non-separating cycles in directed surface graphs, with running times $O(n^2 \log n)$ and $O(g^{1/2} n^{3/2} \log n)$. Erickson and Nayyeri described an algorithm to compute the shortest non-separating cycles in $2^{O(g)} n \log n$ time [29]. Later Fox [33] described algorithms to compute shortest non-contractible cycles in $O(\beta^3 n \log n)$ time and shortest non-separating cycles in $O(\beta^2 n \log n)$ time on surfaces with first Betti number β . (As in previous papers, the input size n is the total number of vertices and edges in the input graph.)

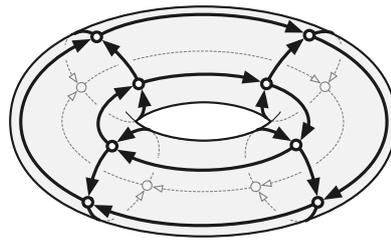
This paper describes the first algorithms and hardness results for finding topologically *trivial* closed walks in directed surface graphs. Our results extend similar results of Cabello, Colin de Verdière, and Lazarus [6, 9] for undirected surface graphs; however, our algorithms require several new techniques, both in design and analysis. (On the other hand, our NP-hardness proofs are actually simpler than the corresponding proofs for undirected graphs!)

We present results for eight different problems, determined by three independent design choices. First, we consider two types of “trivial” closed walks: *contractible* walks, which can be continuously deformed to a point, and *bounding* walks, which are weighted sums of face boundaries. (See Section 2 for more detailed definitions.) Second, like Cabello et al. [6, 8, 9], we carefully distinguish between closed walks and simple cycles throughout the paper. Finally, we consider two different goals: deciding whether a given directed graph contains a trivial cycle or closed walk, and finding the shortest trivial cycle or closed walk in a given directed graph (possibly with weighted edges). Crucially, our algorithms do *not* assume that the faces of the input embedding are open disks. Our results are summarized in Table 1.

■ **Table 1** Our results; β is the first Betti number of the underlying surface.

Structure	Surface	Any	Shortest
Simple contractible cycle	any	$O(n)$	NP-hard
Contractible closed walk	annulus	$O(n)$	$O(n^2 \log \log n)$
	torus	$O(n)$	$O(n^3 \log \log n)$
	boundary	$O(n)$	$O(\beta^5 n^3)$
	other	$O(n)$	$O(\beta^6 n^9)$
Simple bounding cycle	any	NP-hard	NP-hard
Bounding closed walk	any	$O(\beta n)$	NP-hard

In Section 3, we describe linear-time algorithms to determine whether a directed surface graph contains a simple contractible cycle or a contractible closed walk, matching similar algorithms for undirected graphs by Cabello et al. [9]. Our algorithms are elementary: After removing some obviously useless edges, we report success if and only if some face of the input embedding has a (simple) contractible boundary. However, the proofs of correctness require careful analysis of the dual graphs, and the correctness proof for contractible closed walks relies on a subtle topological lemma of Hass and Scott [38]. These two problems are nontrivial for directed graphs *even if* every face of the input embedding is a disk; see Figure 1.



■ **Figure 1** A cellularly embedded directed graph with no contractible or bounding closed walks.

In Section 4, we describe an algorithm to determine whether a directed surface graph contains a *bounding* closed walk in $O(\beta n)$ time. We exploit a careful analysis of the interplay between strong connectivity in the input graph G and its dual graph G^* . With some additional effort, our algorithm can return an explicit description of a bounding closed walk in $O(n^2)$ time if one exists; in the full paper [30, Section 4.2] we prove that this quadratic upper bound is optimal. This problem can be reduced to finding zero cycles in periodic (or “dynamic”) graphs [17, 40, 45]; a periodic graph is a graph whose edges are labeled with integer vectors; a zero cycle is a closed walk whose edge labels sum to the zero vector. However, all algorithms known for finding zero cycles rely on linear programming, and thus are much more complex and much less efficient than the specialized algorithm we present.

Next, we propose three polynomial-time algorithms to compute shortest contractible closed walks. Each of our algorithms is designed for a different class of surfaces, depending whether the surface’s fundamental group is abelian (the annulus and the torus), free (any surface with boundary), or hyperbolic (everything else). Our algorithm for the annulus and torus uses a standard covering-space construction, together with a recent algorithm of Mozes et al. [50]. For graphs on surfaces with boundary, we exploit the fact that the set of trivial words for any finitely-generated free group is a context-free language [51, 54]; this observation allows us to reduce to a small instance of the *CFG shortest path* problem [3, 4, 56]. We describe these two algorithms in detail in the full paper [30, Section 6]. In Section 5, we describe our algorithm for hyperbolic surfaces, which also reduces to CFG shortest paths; the main technical hurdle is the construction of an appropriate context-free grammar. Specifically, for any integers $g \geq 2$ and $L \geq 0$, we construct a context-free grammar with $O(g^2 L^2)$ nonterminals, in Chomsky normal form, that generates all contractible closed walks of length L , and only contractible closed walks, in a canonical genus- g surface map called a *system of quads* [31, 47]. Our construction exploits well-known geometric properties of hyperbolic tilings.

Finally, in the full paper [30, Section 5], we prove that detecting simple bounding cycles, finding shortest simple contractible cycles, and finding shortest bounding closed walks are all NP-hard. Cabello [6] described a polynomial-time algorithm to compute the shortest simple contractible cycle in an *undirected* surface graph; thus, our reduction for that problem makes essential use of the fact that the input graph is directed. Cabello [6] and Cabello et al. [9] proved that the other two problems are NP-hard in undirected surface graphs. Our NP-hardness proofs closely follow theirs but are slightly simpler.

2 Background

Directed graphs. Let G be an arbitrary directed graph, possibly with loops and parallel edges. Each edge of G is directed from one endpoint, called its *tail*, to the other endpoint, called its *head*. An edge is a *loop* if its head and tail coincide. At the risk of confusing the reader, we sometimes write $u \rightarrow v$ to denote an edge with tail u and head v .

34:4 Topologically Trivial Closed Walks in Directed Surface Graphs

A *walk* in G is an alternating sequence of vertices and edges $v_0 \rightarrow v_1 \rightarrow \cdots \rightarrow v_\ell$, where $v_i \rightarrow v_{i+1}$ is an edge in G for each index i ; this walk is *closed* if $v_0 = v_\ell$. A *simple cycle* is a closed walk that visits each vertex at most once. An *edge cut* in a directed graph G is a nonempty subset X of edges such that $G \setminus X$ has two components, one containing the tails of edges in X , and other containing the heads of edges in X . A directed graph is *strongly connected* if it contains a walk from any vertex to any other vertex, or equivalently, if it contains no edge cuts.

An (integer) *circulation* in a directed graph G is a function $\phi: E(G) \rightarrow \mathbb{N}$ that satisfies the balance constraint $\sum_{u \rightarrow v} \phi(u \rightarrow v) = \sum_{v \rightarrow w} \phi(v \rightarrow w)$ for every vertex v . The *support* of ϕ is the subset of edges e such that $\phi(e) > 0$. An *Euler tour* of ϕ is a closed walk that traverses each edge e exactly $\phi(e)$ times; such a walk exists if and only if the support of ϕ is connected.

Surfaces, embeddings, and duality. A *surface* is a 2-manifold, possibly with boundary. A surface is *orientable* if it does not contain a Möbius band; we explicitly consider only orientable surfaces in this paper. A *closed curve* on a surface S is (the image of) a continuous map $\gamma: S^1 \hookrightarrow S$; a closed curve is *simple* if this map is injective. The *boundary* ∂S of S consists of disjoint simple closed curves; the *interior* of S is the complement $S^\circ = S \setminus \partial S$. The *genus* of S is the maximum number of disjoint simple closed curves in S° whose deletion leaves the surface connected. Up to homeomorphism, there is exactly one orientable surface with genus g and b boundary cycles, for any non-negative integers g and b . The *first Betti number* of an orientable surface is either $2g$ if $b = 0$, or $2g + b - 1$ if $b > 0$.

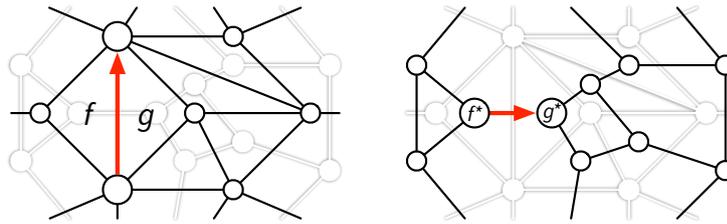
An *embedding* of G on a surface S is a continuous map that sends vertices of G to distinct points in S° , and sends edges to interior-disjoint simple paths in S° from their tails to their heads. In particular, if G contains two anti-parallel edges $u \rightarrow v$ and its reversal $v \rightarrow u$, those edges are embedded as interior-disjoint paths. The embedding of G maps every (simple) closed walk in G to a (simple) closed curve in S° ; we usually do not distinguish between a closed walk in G and its image in S .

We explicitly consider graphs with loops and parallel edges; however, without loss of generality, we assume that no loop edge is the boundary of a disk, and no two parallel edges are the boundary of a disk. With this assumption, Euler's formula implies that a graph with n vertices on a surface with first Betti number β has at most $O(n + \beta)$ edges.

A *face* of an embedding is a component of the complement of the image of the graph. An embedding is *cellular* if every face is homeomorphic to an open disk. *Unlike most previous papers, we explicitly consider non-cellular graph embeddings*; a single face can have disconnected boundary and/or positive genus. Each directed edge e in a surface graph lies on the boundary of two (possibly equal) faces, called the *left shore* and *right shore* of e . We sometimes write $f \uparrow f'$ to denote a directed edge whose left shore is f and whose right shore is f' . A *boundary face* of the embedding is any face that intersects the boundary of S .

A *dual walk* is an alternating sequence of faces and edges $f_0 \uparrow f_1 \uparrow \cdots \uparrow f_\ell$, where $f_i \uparrow f_{i+1}$ is an edge in G for each index i . A dual walk is *closed* if its initial face f_0 and final face f_ℓ coincide; a dual walk is *simple* if the faces f_i are distinct (except possibly $f_0 = f_\ell$). A simple closed dual walk is called a *cocycle*. A minimal edge cut in a surface graph is the disjoint union of at most $g + 1$ cocycles.

Every surface embedding of a directed graph G defines a directed *dual graph* G^* , with one vertex f^* for each face f of G , and one edge e^* for each edge e of G . Specifically, for any edge e in G , we have $\text{tail}(e^*) = \text{left}(e)^*$ and $\text{head}(e^*) = \text{right}(e)^*$, as shown in Figure 2. We will treat G^* exclusively as an abstract directed graph. Every dual walk in G corresponds to a walk in G^* ; in particular, every cocycle in G corresponds to a cycle in G^* .



■ **Figure 2** Duality in directed surface graphs.

Contractible and bounding. Let $\alpha: [0, 1] \rightarrow S$ and $\beta: [0, 1] \rightarrow S$ be two (not necessarily simple) paths in S with the same endpoints. A *homotopy* between α and β is a continuous function $h: [0, 1]^2 \rightarrow S$ such that $h(s, 0) = \alpha(0) = \beta(0)$ and $h(s, 1) = \alpha(1) = \beta(1)$ for all s , and $h(0, t) = \alpha(t)$ and $h(1, t) = \beta(t)$ for all t . Two paths are *homotopic*, or in the same *homotopy class*, if there is a homotopy between them. A closed curve γ in S (or a closed walk in a surface graph G) is *contractible* if it is homotopic to a constant function. A simple closed curve (or a simple cycle in G) is contractible in S if and only if it is the boundary of a disk in S [25].

Mirroring classical terminology for curves in the plane [1,52], an *Alexander numbering* for a surface graph G is a function $\alpha: F(G) \rightarrow \mathbb{Z}$ that assigns an integer to each face of G , such that $\alpha(f) = 0$ for every boundary face f . The *boundary* $\partial\alpha$ of an Alexander numbering α is a circulation $\partial\alpha: E(G) \rightarrow \mathbb{Z}$, defined by setting $\partial\alpha(e) = \alpha(\text{left}(e)) - \alpha(\text{right}(e))$. A closed walk is *bounding* if and only if it is an Euler tour of some boundary circulation. Equivalently, a closed walk (or its underlying circulation) is bounding if and only if its *integer* homology class is trivial. (We refer to reader to Hatcher [39], Giblin [35], or Edelsbrunner and Harer [23] for a more through introduction to homology.) On the sphere or the plane, every closed walk is bounding, and every circulation is a boundary circulation; however, these equivalences do not extend to other surfaces. (See Figure 1!) A simple cycle γ in G is bounding in S if and only if $S \setminus \gamma$ is disconnected; simple bounding cycles are often called *separating* cycles.

3 Contractible Cycles and Walks

3.1 Simple Cycles

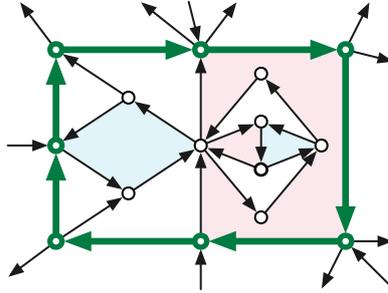
► **Lemma 1.** *Let G be a directed graph embedded on an orientable surface, and let e be a directed edge that lies in a directed cycle of G . No bounding closed walk (and in particular, no simple contractible cycle) in G traverses e .*

Proof. Let λ be a directed cocycle in G , and let ω be a bounding closed walk. Let $\phi: E(G) \rightarrow \mathbb{N}$ be the boundary circulation defined by setting $\phi(e)$ to the number of times that ω traverses e , and let $\alpha: F(G) \rightarrow \mathbb{N}$ be an Alexander numbering of ω (so that $\phi = \partial\alpha$). We immediately have $\sum_{e \in \lambda} \phi(e) = \sum_{e \in \lambda} (\alpha(\text{right}(e)) - \alpha(\text{left}(e))) = 0$. Because $\phi(e) \geq 0$ for every edge e in G , it follows that $\phi(e) = 0$ for every edge $e \in \lambda$. ◀

In light of this lemma, we can assume without loss of generality that G contains no directed cocycles; in particular, no edge of G has the same face on both sides.

The *boundary* of a face f , denoted ∂f , is the set of edges that have f on one side. The boundary of f is oriented *clockwise* if f is the right shore of any edge in ∂f , and *counterclockwise* if f is the left shore of any edge in ∂f . Each face in G with counterclockwise

boundary appears as a source in the dual graph G^* ; each face of G with clockwise boundary appears as a sink in G^* . We call a face *coherent* if its boundary is oriented either clockwise or counterclockwise, and *incoherent* otherwise. See Figure 3.



■ **Figure 3** A simple clockwise contractible cycle (bold green) in a directed surface graph, enclosing four coherent faces: two counterclockwise (shaded blue) and two clockwise (shaded pink).

► **Lemma 2.** *Let G be a directed graph with no cocycles, embedded on an orientable surface S . If G contains a simple contractible directed cycle, then G has a face whose boundary is a simple contractible directed cycle.*

Proof. Let γ be a simple contractible cycle in G . This cycle is the boundary of a closed disk D [25]. Without loss of generality, assume γ is oriented clockwise around D , so the right shore of every edge in γ is a face in D .

A dual walk that starts at a face inside D cannot visit the same face more than once, because G has no cocycles (G^* has no cycles), and cannot cross γ , because γ is oriented clockwise (all its dual edges point into D). Thus, D must contain at least one face f with clockwise boundary (a sink in G^*). The closure of f is a subset of the closed disk D , so it must be homeomorphic to a closed disk with zero or more interior holes.

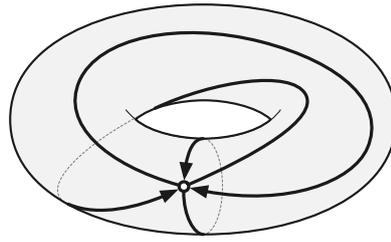
If ∂f is a simple directed cycle, we are done. Otherwise, ∂f is the union of two or more edge-disjoint simple directed cycles; consider the non-simple shaded face in Figure 3. Each of these cycles is contractible, because it lies in D . Any simple cycle in ∂f that is not the original cycle γ (for example, the boundary of any hole in the closure of f) encloses strictly fewer faces than γ . The lemma now follows immediately by induction on the number of enclosed faces. ◀

► **Theorem 3.** *Given an arbitrary directed graph G embedded on an arbitrary orientable surface, we can determine in $O(n)$ time whether G contains a simple contractible cycle.*

Proof. The algorithm proceeds as follows. First, we compute the strong components of the dual graph G^* in $O(n)$ time. An edge e of G lies in a cocycle if and only if both endpoints of the dual edge e^* lie in the same strong component of G^* ; we remove all such edges from G in $O(n)$ time. Finally, we examine each face of the remaining subgraph by brute force, in $O(n)$ time. If any face is an open disk whose boundary is a simple directed cycle, we output TRUE; otherwise, we output FALSE. Correctness follows from Lemmas 1 and 2. ◀

3.2 Non-simple Closed Walks

It is easy to construct directed surface graphs that contain contractible closed walks but no simple contractible cycles. For example, the one-vertex graph in Figure 4 has two Eulerian circuits, both of which are contractible, but the only simple cycles consist of single edges, none of which are contractible.



■ **Figure 4** A surface graph with contractible closed walks but no simple contractible cycles.

We detect contractible closed walks using essentially the algorithm described in the previous section: After removing all cocycle edges, we look for a face whose boundary is coherent and whose interior is an open disk. However, proving this algorithm correct requires more subtlety. Here we only sketch a proof of correctness, deferring full details to the full version [30, Section 3.2].

First, we observe that some shortest contractible closed walk in G is *weakly simple*, meaning it can be perturbed in a small neighborhood of the graph into a simple closed curve [15]. Let ω be a shortest contractible closed walk in G , and let $\tilde{\omega}$ be a generic perturbation of ω on the surface. A seminal result of Hass and Scott [38, Theorem 2.7] implies that if $\tilde{\omega}$ is not already simple, then it contains either a simple contractible closed subpath, which we call a *monogon*, or a pair of simple interior-disjoint subpaths that bound a disk, which we call a *bigon*. We call a bigon *coherent* if one subpath is homotopic to the reverse of the other, and *incoherent* otherwise. See Figure 5. If $\tilde{\omega}$ contained a monogon or a coherent bigon, smoothing its endpoint(s) would give us shorter contractible closed walk than ω , which is impossible. Smoothing the corners of incoherent bigons removes all self-intersections from $\tilde{\omega}$; we can mirror this smoothing by reordering subpaths of the walk ω .



■ **Figure 5** Shortening or simplifying a contractible closed curve by smoothing at points of self-intersection. From left to right: A monogon, a coherent bigon, and an incoherent bigon.

A simple winding-number argument implies that any weakly simple *bounding* walk in G traverses each edge at most once.

Finally, we prove that if a graph G with no cocycles has a contractible closed walk, then the boundary of some face of G is contractible, using a similar induction argument as Lemma 2. If some weakly simple contractible closed walk ω encloses more than one face, then it must enclose a face f with coherent boundary (a source or sink in G^*). If face f is a disk, we are done; otherwise, f is a disk with holes, and the boundary of any hole is another weakly simple contractible closed walk that encloses fewer faces.

► **Theorem 4.** *Given a directed graph G embedded on any surface, we can determine in $O(n)$ time whether there is a contractible closed walk in G .*

If the face f has an interior hole, then the graph G *must* be disconnected, because the interior of f is a subset of the open disk A . Thus, it may tempting to simplify our argument by assuming “without loss of generality” that the graph G is connected. Unfortunately, we cannot *simultaneously* assume that G is connected *and* that G contains no cocycles. Resolving this tension lies at the core of our algorithm for finding bounding closed walks, which we describe next.

4 Bounding Closed Walks

To simplify our presentation, we assume in this section that our input graphs are embedded on surfaces *without* boundary; this assumption is justified by the following observation. Let G be a directed graph embedded on a surface S with $b > 0$ boundary cycles, and let S^\bullet be the surface without boundary obtained from S by attaching a single disk with $b - 1$ holes to all boundary cycles of S . All boundary faces of G on the original surface S are contained in a single face of G on S^\bullet . If S has first Betti number β , then S^\bullet has genus $O(\beta)$.

► **Lemma 5.** *A closed walk ω in G is bounding in S if and only if it is bounding in S^\bullet .*

Proof. Any Alexander numbering for ω on S is also an Alexander numbering for ω on S^\bullet .

Let α be any Alexander numbering for ω on S^\bullet . For any integer k , the function $\alpha + k$ defined as $(\alpha + k)(f) = \alpha(f) + k$ for every face is also an Alexander numbering for ω on S^\bullet . Thus, there is an Alexander numbering α^\bullet of ω such that $\alpha^\bullet(f) = 0$ for the single face of G containing $S^\bullet \setminus S$. The function α^\bullet is an Alexander numbering for ω on S . ◀

Our algorithm for detecting bounding closed walks relies on two elementary observations. First, we can assume without loss of generality that G is strongly connected, because any closed walk in the input graph G stays within a single strong component of G . On the other hand, Lemma 1 implies that we can assume without loss of generality that the input graph G has no cocycles; otherwise, we can remove all cocycles in $O(n)$ time by contracting each strong component of the dual graph G^* to a single dual vertex. Indeed, if the input graph satisfies *both* of these conditions, finding bounding closed walks is trivial.

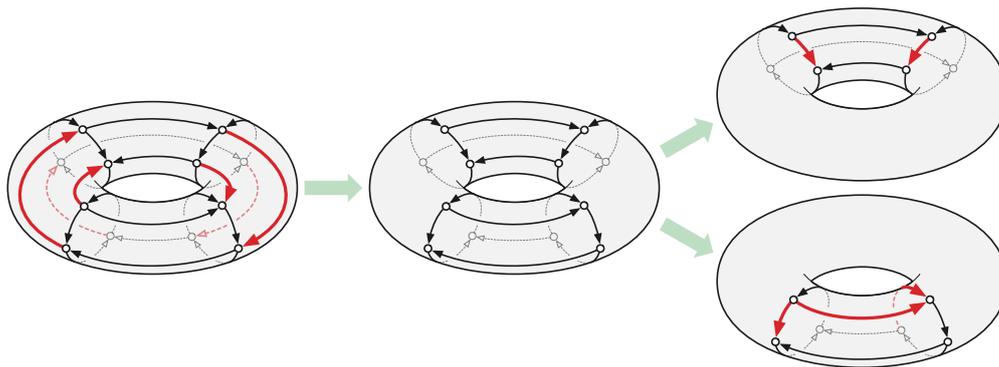
► **Lemma 6.** *Let G be a strongly connected surface graph with at least one edge, whose dual graph G^* is acyclic. There is a bounding closed walk in G .*

Proof. For each face f of G , let $\alpha(f)$ be the rank of the dual vertex f^* in an arbitrary topological sort of the dual graph G^* . For each edge e in G , we have $\alpha(\text{left}(e)) > \alpha(\text{right}(e))$. Thus, the function $\partial\alpha: e \mapsto \alpha(\text{left}(e)) - \alpha(\text{right}(e))$ is a positive integer boundary circulation with connected support. Any Euler tour of this circulation is a bounding closed walk. ◀

Kao and Shannon [41, 43, 44] observed that these two assumptions are actually identical for planar graphs; a planar directed graph G is strongly connected if and only if its dual graph G^* is acyclic. (This important observation is the basis of several efficient algorithms for planar directed graphs [2, 41–44, 53].) And indeed, finding bounding closed walks in planar graphs is trivial, because *every* closed walk in a planar graph is bounding.

However, this equivalence does not extend to directed graphs on more complex surfaces; a strongly connected directed surface graph can contain many directed cocycles. Moreover, deleting all the directed cocycles from a strongly connected surface graph can disconnect the graph. More subtly, if H is a disconnected surface graph without cocycles, the induced embeddings of the individual components of H *can* contain cocycles. See Figure 6.

Our algorithm repeatedly applies both of these simplifications, alternately removing cocycle edges and separating components, eventually reporting success if it ever finds a component that is both strongly connected and cocycle-free. Each iteration can be performed in $O(n)$ time, and each iteration removes at least one edge, so conservatively, our algorithm runs in $O(n^2)$ time. But in fact, our algorithm converges after only $O(g)$ iterations. We only sketch a proof of this claim in this extended abstract; see the full paper [30, Section 4.1] for complete proofs.



■ **Figure 6** A strongly connected directed graph on the torus with cocycles (bold red). Deleting all cocycles disconnects the graph. The components of the disconnected graph have more cocycles. (Only one cocycle is emphasized in each component.)

► **Lemma 7.** *Let H be a directed surface graph without cocycles. Every weak component of H is strongly connected.*

Proof. If a weak component of H is not strongly connected, then H contains a non-empty directed edge cut. Every non-empty directed edge cut is the disjoint union of cocycles. ◀

We call any face of G *simple* if it is homeomorphic to an open disk and *non-simple* otherwise. (The boundary of a simple face is *not* necessarily a simple cycle.)

► **Lemma 8.** *Let G be a strongly connected surface graph, let H be the subgraph of G obtained by deleting all cocycles, and let G' be a (strong) component of H . Every simple face of G' is also a simple face of G .*

► **Corollary 9.** *Let G be a strongly connected surface graph, let H be the subgraph of G obtained by deleting all cocycles, and let G' be a (strong) component of H . Every directed cocycle in G' visits at least one non-simple face of G' .*

We define the *footprint* $\llbracket G \rrbracket$ of a directed surface graph G as the union of the vertices, edges, and *simple* faces of G . For example, the footprint of the middle graph in Figure 6 is the disjoint union of two annuli. The embedding of G is cellular if and only if $\llbracket G \rrbracket$ is the entire surface. Let $\beta_0(G)$ denote the number of (weak) components of G , or equivalently, the number of components of its footprint $\llbracket G \rrbracket$. Let $\beta_1(G)$ denote the first Betti number of $\llbracket G \rrbracket$, which is the rank of the first homology group of $\llbracket G \rrbracket$. If G has at least one non-simple face, then $\beta_1(G) := \beta_0(G) - v(G) + e(G) - f_0(G)$, where $v(G)$, $e(G)$, and $f_0(G)$ denote the number of vertices, edges, and simple faces of G , respectively. If G (and therefore $\llbracket G \rrbracket$) is disconnected, then $\beta_1(G)$ is the sum of the first Betti numbers of its components.

► **Lemma 10.** *Let G be a strongly connected surface graph, let H be the subgraph of G obtained by deleting all cocycles, let G' be a (strong) component of H , and let H' be the subgraph of G' obtained by deleting all cocycles. If $H' \neq G'$, then $\beta_1(H') < \beta_1(G')$.*

► **Theorem 11.** *Given a directed graph G embedded on any surface with genus g , we can determine in $O(gn)$ time whether there is a bounding closed walk in G .*

Proof. Assume G is strongly connected, since otherwise, we can consider each strong component of G separately. Let H be the graph obtained by deleting all cocycles of G . We can construct H in $O(n)$ time by computing the strong components of G^* in linear time, and

then deleting any edge of G whose incident faces lie in the same strong component of G^* . Lemma 1 implies that any bounding closed walk in G is also a bounding closed walk in H .

If H has no edges, we return FALSE. If H is weakly connected and has at least one edge, then, following Lemmas 6 and 7, we immediately return TRUE. Otherwise, we recursively examine each (strong) component of H . Each vertex of the input graph G participates in only one subproblem at each level of recursion, so the total time spent at each level is $O(n)$. Finally, Lemma 10 implies that the depth of the recursion tree is at most $O(g)$. ◀

5 Shortest Contractible Closed Walks

Finally, we describe polynomial-time algorithms to compute the shortest contractible closed walk in a directed surface graph with weighted edges. In this extended abstract, we describe only our algorithm for surfaces without boundary and with genus $g \geq 2$. These are precisely the surfaces whose fundamental groups are *hyperbolic* in the sense of Gromov [37], but not free; as already observed by Dehn [19], the natural geometry for the universal cover of these surfaces is the hyperbolic plane. (See Figure 8 below.) We describe our (simpler) algorithms for other types of surfaces in the full paper [30, Section 6].

Ultimately, our algorithm reduces the problem of finding the shortest contractible cycle to the *CFG shortest path* problem, introduced by Yannakakis [56]. Given a directed graph $G = (V, E)$ with weighted edges, a labeling $\ell: E \rightarrow \Sigma \cup \{\varepsilon\}$ for some finite set Σ , and a context-free grammar \mathcal{G} over the alphabet Σ , the CFG shortest path problem asks for the shortest walk (if any) in G whose label is in the language generated by \mathcal{G} . An algorithm of Barrett et al. [3] solves the CFG shortest path problem in $O(NPn^3)$ time, where N and P are the numbers of nonterminals and productions in \mathcal{G} , respectively, and n is the number of vertices in G , when the grammar \mathcal{G} is in Chomsky normal form and all edge weights in G are non-negative. (Their algorithm was extended to graphs with negative edges but no negative cycles by Bradford and Thomas [4].) The main difficulty in our algorithm is the construction of an appropriate context-free grammar.

Fix a directed graph G with non-negatively weighted edges, embedded on an orientable surface S with genus $g \geq 2$ and no boundary. Unlike previous sections, it will prove convenient in this section to treat G as undirected, but with asymmetric edge weights, as proposed by Cabello et al. [10]. That is, each undirected edge uv in G is composed of two *darts* $u \rightarrow v$ and $v \rightarrow u$ with independent weights; a walk is alternating series of vertices and *darts*; and the length of a walk is the sum of the weights of its darts, counted with appropriate multiplicity. We also assume without loss of generality that the given embedding of G is *cellular*, meaning every face is homeomorphic to an open disk. These assumptions can be enforced if necessary by adding at most $O(m)$ directed edges with (symbolically) infinite weight.

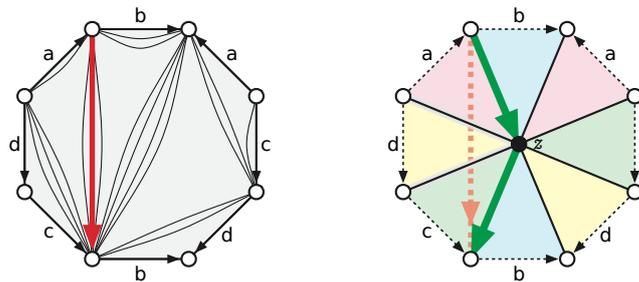
5.1 Edge Labeling

The first step in our algorithm is to label the edges of G so that the labels along any closed walk encode its homotopy type. In principle, we can derive such a labeling by reducing G to a system of loops [12, 22, 46]; however, this labeling leads to a less efficient algorithm.¹ Instead, we reduce G to a different canonical surface decomposition called a *system of quads*, as proposed by Lazarus and Rivaud [47] and Erickson and Whittlesey [31].

¹ Specifically, the running time of the resulting algorithm has an extra factor of $g^{O(g)}$ in the running time.

Let (T, L, C) be an arbitrary *tree-cotree decomposition* of G , which partitions the edges of G into a spanning tree T , a spanning tree C^* of the dual graph G^* , and exactly $2g$ leftover edges L [24]. Contracting every edge in T and deleting every edge in C reduces G to a *system of loops*, which has one vertex a , one face f , and $2g$ loops L . To construct the system of quads Q , we introduce a new vertex z in the interior of f , add edges between z and every corner of f , and then delete the edges in L .

Next we label each directed edge e in G with a directed walk $\langle e \rangle$ in Q as follows. Every walk $\langle e \rangle$ starts and ends at a and thus has even length. If $e \in T$, then $\langle e \rangle$ is the empty walk. Otherwise, after contracting T , edge e connects two corners of f ; we define $\langle e \rangle$ as the walk of length 2 in Q from the back corner of e , to z , and then to the front corner of e . (If $e \in L$, there are two such walks; choose one arbitrarily.)



■ **Figure 7** Left: A fundamental polygon for a surface of genus 2, obtained by contracting a spanning tree and cutting along a system of loops. Right: Mapping a directed non-tree edge to a path of length 2 in the corresponding system of quads.

Finally, for any closed walk ω in G , let $\langle \omega \rangle$ denote the closed walk in Q obtained by concatenating of the labels of edges in ω in order. The closed walks ω and $\langle \omega \rangle$ are freely homotopic in S ; in particular, ω is contractible if and only if $\langle \omega \rangle$ is contractible. Moreover, the number of edges in $\langle \omega \rangle$ is at most twice the number of edges in ω .

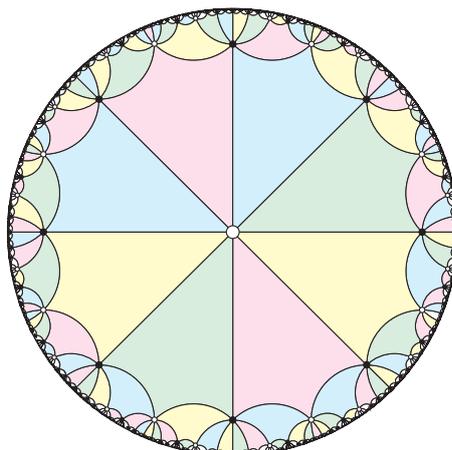
The universal cover \widehat{Q} of a system of quads Q is isomorphic to a regular tiling of the hyperbolic plane by quadrilaterals meeting at vertices of degree $4g$; see Figure 8. The covering map from \widehat{Q} to Q is a graph homomorphism (mapping vertices to vertices and edges to edges), which (at the risk of confusing the reader) we also denote $\langle \cdot \rangle: \widehat{Q} \rightarrow Q$. A closed walk in Q is contractible if and only if it is the projection of a closed walk in \widehat{Q} . Thus, a closed walk ω in G is contractible if and only if there is a closed walk $\widehat{\omega}$ in \widehat{Q} such that $\langle \widehat{\omega} \rangle = \langle \omega \rangle$.

5.2 Hyperbolic Geometry

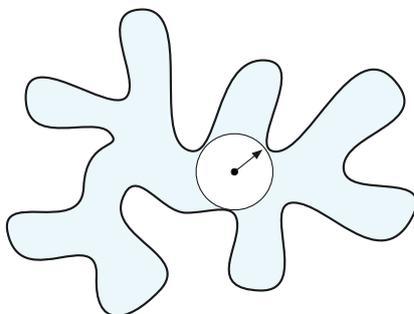
Our algorithm exploits two well-known geometric properties of regular hyperbolic tilings: the area of any ball in a hyperbolic tiling grows exponentially with its radius [36, 48], and the area enclosed by any simple cycle grows at most linearly with the cycle's length [20, 21]. In particular, we rely on the following immediate corollary of these two properties: The distance from any point in the interior of a simple cycle in a regular hyperbolic tiling is at most *logarithmic* in the length of that cycle; see Figure 9.

Classical results already imply this *qualitative* behavior for *all* regular hyperbolic tilings; indeed, Chepoi et al. [16] have derived tight *asymptotic* bounds. However, we require *precise* upper bounds for growth, isoperimetry, and in-radius in the specific tiling \widehat{Q} ; the running time of our algorithm ultimately depends *exponentially* on the constants in those bounds.

For any positive integer r , let $N(r)$ denote the number of vertices at distance at most r from an arbitrary *basepoint* vertex \widehat{a} of \widehat{Q} ; we derive tight bounds on $N(r)$ from a recurrence



■ **Figure 8** The universal cover of a genus-2 system of quads. (Compare with Figure 7.)



■ **Figure 9** Simple closed curves in hyperbolic tilings have logarithmic in-radius.

of Floyd and Plotnick [32]. We derive our isoperimetric inequalities using the “spur and bracket” analysis of Erickson and Whittlesey [31, Section 4.2], which is based in turn on results of Gersten and Short [34], along with classical results of Dehn [20, 21]. Our inradius bound then follows immediately. See the full paper [30, Section 7.2] for further details.

► **Lemma 12.** $N(r) \sim \beta \lambda^r$ for some constants $1 < \beta < 2$ and $4g - 3 < \lambda < 4g - 2$. In particular, $N(r) \geq \lambda^r$ for all $r \geq 0$.

► **Lemma 13.** Any simple cycle of length L in \widehat{Q} has less than $3L/2$ faces of \widehat{Q} in its interior.

► **Lemma 14.** Any simple cycle of length L in \widehat{Q} has less than $2L/g$ vertices of \widehat{Q} in its interior.

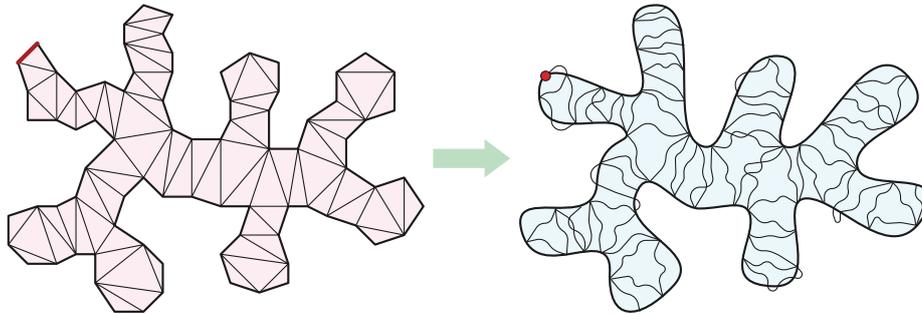
► **Lemma 15.** Let γ be a simple cycle in \widehat{Q} with length at most L . Every tiling vertex in the interior of γ has distance at most $\rho = \lceil \log_\lambda(2L/g) \rceil$ from at least one vertex of γ .

5.3 Triangulation

Now let ω be a closed walk in \widehat{Q} with a distinguished vertex \hat{a} called its *basepoint*. Following Muller and Schupp [51], define a *k-triangulation* of ω is a continuous map \mathcal{T} from a *reference* triangulation T of a simple polygon P to \widehat{Q} with the following properties:

- \mathcal{T} maps vertices of P to vertices of ω .
- \mathcal{T} maps a distinguished edge of P , called the *root edge*, to the basepoint \hat{a} .

- \mathcal{T} maps the boundary of P continuously onto the walk ω ; in particular, every edge of P except the root edge is mapped to a single edge of ω .
 - Finally, \mathcal{T} maps each diagonal edge in T to a walk of length at most k in \widehat{Q} .
- This map is not necessarily an embedding, even locally. The diagonal walks may intersect each other or ω , even if ω is a simple cycle; moreover, some diagonal walks may have length zero if ω is not simple.

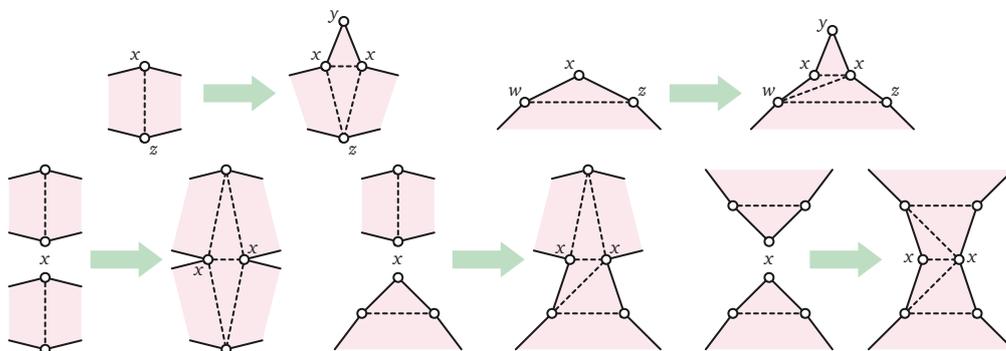


■ **Figure 10** Any closed walk in a hyperbolic tiling can be “triangulated” by paths of logarithmic length.

► **Lemma 16.** *Every simple cycle of length L in \widehat{Q} has a $(2\rho + 2)$ -triangulation, where $\rho = \lceil \log_\lambda(2L/g) \rceil$.*

Proof. Let γ be a simple cycle of length L in \widehat{Q} , and let Γ denote the closed disk bounded by γ . Intuitively, we construct a “Delaunay triangulation” of the vertices of γ inside Γ .

For each tiling vertex x in Γ , let $nn(x)$ denote its nearest neighbor on γ , measuring distance by counting edges in \widehat{Q} and breaking ties arbitrarily. In particular, if $x \in \gamma$, then $nn(x) = x$. Lemma 15 implies that the distance from any vertex x to its nearest neighbor $nn(x)$ is at most ρ . Arbitrarily triangulate each tile in Γ to obtain a triangulation Δ . Let M denote the set of all edges xy in Δ such that $nn(x) \neq nn(y)$. (Intuitively, M is a “discrete medial axis” of Γ [5].) For each edge xy in M , let $\sigma(x, y)$ denote a shortest path in \widehat{Q} from $nn(x)$ to $nn(y)$; in particular, if xy is an edge of γ , then $\sigma(x, y) = xy$. The triangle inequality implies that each shortest path $\sigma(x, y)$ has length at most $2\rho + 2$. The shortest paths $\{\sigma(x, y) \mid xy \in M\}$ are the paths of a $(2\rho + 2)$ -triangulation of γ . ◀



■ **Figure 11** Top: Extending a k -triangulation across a spur. Bottom: Merging two k -triangulations across a generic repeated vertex.

We extend the previous construction to non-simple closed walks using a straightforward induction argument. Any non-simple closed walk can be decomposed into two shorter closed walks at a repeated vertex. The induction hypothesis gives us two $(2\rho + 2)$ -triangulations for the shorter closed walks, which we can extend or combine into a $(2\rho + 2)$ -triangulation for ω by case analysis, as suggested by Figure 11. We defer further details to the full paper [30, Section 7.3].

► **Lemma 17.** *Every closed walk of length L in \widehat{Q} has a $(2\rho + 2)$ -triangulation, where $\rho = \lceil \log_\lambda(2L/g) \rceil$.*

5.4 Context-Free Grammar

Now finally we reach the key component of our algorithm: the construction of a small context-free grammar \mathcal{G} in Chomsky normal form such that the shortest contractible closed walk in G is also the shortest walk whose label is in the language generated by \mathcal{G} .

► **Lemma 18.** *For any positive integer L , there is a context-free grammar \mathcal{G} in Chomsky normal form with $O(g^2L^2)$ non-terminals, such that (1) Every string generated by \mathcal{G} describes a contractible closed walk in Q , and (2) every contractible closed walk in Q of length at most L is generated by \mathcal{G} .*

Proof sketch. We closely follow a construction of Muller and Schupp [51, Theorem 1]. The alphabet for our grammar consists of all $8g$ darts in Q . The variables of \mathcal{G} correspond to homotopy classes of walks of length at most $2\rho + 2$ in Q ; in particular, the starting variable corresponds to the class of contractible walks from vertex a to vertex a . The number of nonterminals is

$$2N(2\rho + 2) = O(\lambda^{2\rho+2}) \leq O(\lambda^{2\log_\lambda(2L/g)+4}) = O((2L/g)^2 \lambda^4) = O(g^2L^2)$$

by Lemma 12 and the definition of ρ . The non-terminal productions in \mathcal{G} correspond to concatenation of homotopy classes; crudely, there are at most $O(g^4L^4)$ such productions. Finally, the grammar includes a terminal production for the homotopy class of each dart. For any closed walk ω of length L , and for any $(2\rho+2)$ -triangulation \mathcal{T} of ω , the dual tree of \mathcal{T} is a parse tree for ω in this grammar. We defer further details to the full paper [30, Section 7.4]. ◀

► **Theorem 19.** *Let G be a directed graph with m non-negatively weighted edges, embedded on the orientable surface of genus g with no boundary. We can compute the shortest contractible closed walk in G in $O(g^6m^9)$ time.*

Proof. We construct the system of quads Q and label the edges of G in $O(m)$ time. If necessary, subdivide edges of G so that each edge is labeled with either the empty walk or a single dart in Q . After subdivision, G has at most $2m$ edges, and therefore (because G is symmetric) at most $2m$ vertices.

Then we build a context-free grammar \mathcal{G} that generates only contractible walks in Q and generates all non-empty contractible walks in Q of length at most $2m$, as described in Lemma 18. This grammar has $N = O(g^2m^2)$ nonterminals and $P = O(g^4m^4)$ productions. Recall from Section 3.2 that some shortest contractible walk in G is weakly simple, and therefore has at most $2m$ edges. It follows that \mathcal{G} generates the label of some shortest contractible walk in G .

Finally, we compute the shortest closed walk in G whose label is generated by \mathcal{G} , using the CFG-shortest-path algorithm of Barrett et al. [3], in $O(NP(n')^3) = O(g^6m^9)$ time. ◀

References

- 1 James W. Alexander. Topological invariants of knots and links. *Trans. Amer. Math. Soc.*, 30(2):275–306, 1928. doi:10.1090/S0002-9947-1928-1501429-1.
- 2 Lars Arge, Laura Toma, and Norbert Zeh. I/O-efficient topological sorting of planar DAGs. In *Proc. 15th Ann. ACM Symp. Parallel Algorithms Arch.*, pages 85–93, 2003. doi:10.1145/777412.777427.
- 3 Chris Barrett, Riko Jacob, and Madhav Marathe. Formal-language-constrained path problems. *SIAM J. Comput.*, 30(3):809–837, 2000. doi:10.1137/S0097539798337716.
- 4 Phillip G. Bradford and David A. Thomas. Labeled shortest paths in digraphs with negative and positive edge weights. *RAIRO-Theor. Inf. Appl.*, 43(3):567–583, 2009. doi:10.1051/ita/2009011.
- 5 Jonathan W. Brandt. Convergence and continuity criteria for discrete approximations of the continuous planar skeletons. *CVGIP: Image Understanding*, 59(1):116–124, 1994. doi:10.1006/ciun.1994.1007.
- 6 Sergio Cabello. Finding shortest contractible and shortest separating cycles in embedded graphs. *ACM Trans. Algorithms*, 6(2):24:1–24:18, 2010. doi:10.1145/1721837.1721840.
- 7 Sergio Cabello, Erin W. Chambers, and Jeff Erickson. Multiple-Source Shortest Paths in Embedded Graphs. *SIAM J. Comput.*, 42(4):1542–1571, 2013. doi:10.1137/120864271.
- 8 Sergio Cabello, Éric Colin de Verdière, and Francis Lazarus. Finding shortest non-trivial cycles in directed graphs on surfaces. In *Proc. 26th Ann. Symp. Comput. Geom.*, pages 156–165, 2010. doi:10.1145/1810959.1810988.
- 9 Sergio Cabello, Éric Colin de Verdière, and Francis Lazarus. Finding cycles with topological properties in embedded graphs. *SIAM J. Discrete Math.*, 25:1600–1614, 2011. doi:10.1137/100810794.
- 10 Sergio Cabello, Éric Colin de Verdière, and Francis Lazarus. Finding shortest non-trivial cycles in directed graphs on surfaces. *J. Comput. Geom.*, 7(1):123–148, 2016. doi:10.20382/jocg.v7i1a7.
- 11 Sergio Cabello, Matt DeVos, Jeff Erickson, and Bojan Mohar. Finding one tight cycle. *ACM Trans. Algorithms*, 6(4):article 61, 2010. doi:10.1145/1824777.1824781.
- 12 Sergio Cabello and Bojan Mohar. Finding Shortest Non-Separating and Non-Contractible Cycles for Topologically Embedded Graphs. *Discrete Comput. Geom.*, 37(2):213–235, 2007. doi:10.1007/s00454-006-1292-5.
- 13 Erin W. Chambers, Éric Colin de Verdière, Jeff Erickson, Francis Lazarus, and Kim Whittlesey. Splitting (complicated) surfaces is hard. *Comput. Geom. Theory Appl.*, 41(1–2):94–110, 2008. doi:10.1016/j.comgeo.2007.10.010.
- 14 Erin W. Chambers, Jeff Erickson, and Amir Nayyeri. Minimum cuts and shortest homologous cycles. In *Proc. 25th Ann. Symp. Comput. Geom.*, pages 377–385, 2009. doi:10.1145/1542362.1542426.
- 15 Hsien-Chih Chang, Jeff Erickson, and Chao Xu. Detecting weakly simple polygons. In *Proc. 26th ACM-SIAM Symp. Discrete Algorithms*, pages 1655–1670, 2015. doi:10.1137/1.9781611973730.110.
- 16 Victor Chepoi, Feodor Dragan, Bertrand Estellon, Michel Habib, and Yann Vaxès. Diameters, centers, and approximating trees of δ -hyperbolic geodesic spaces and graphs. In *Proc. 24th Ann. Symp. Comput. Geom.*, pages 59–68, 2008. doi:10.1145/1377676.1377687.
- 17 Edith Cohen and Nimrod Megiddo. Strongly polynomial-time and NC algorithms for detecting cycles in periodic graphs. *J. Assoc. Comput. Mach.*, 40(4):791–830, 1993. doi:10.1145/153724.153727.
- 18 Éric Colin de Verdière and Jeff Erickson. Tightening non-simple paths and cycles on surfaces. *SIAM J. Comput.*, 39(8):3784–3813, 2010. doi:10.1137/090761653.
- 19 Max Dehn. Über unendliche diskontinuierliche Gruppen. *Math. Ann.*, 71(1):116–144, 1911. URL: <http://eudml.org/doc/158521>.

- 20 Max Dehn. Transformation der Kurven auf zweiseitigen Flächen. *Math. Ann.*, 72(3):413–421, 1912. doi:10.1007/BF01456725.
- 21 Max Dehn. *Papers on Group Theory and Topology*. Springer, 1987. Translated by John Stillwell. doi:10.1007/978-1-4612-4668-8.
- 22 Tamal K. Dey and Sumanta Guha. Transforming Curves on Surfaces. *J. Comput. System Sci.*, 58:297–325, 1999. doi:10.1006/jcss.1998.1619.
- 23 Herbert Edelsbrunner and John L. Harer. *Computational Topology: An Introduction*. Amer. Math. Soc., 2010.
- 24 David Eppstein. Dynamic generators of topologically embedded graphs. In *Proc. 14th Ann. ACM-SIAM Symp. Discrete Algorithms*, pages 599–608, 2003.
- 25 David B. A. Epstein. Curves on 2-manifolds and isotopies. *Acta Mathematica*, 115:83–107, 1966. doi:10.1007/BF02392203.
- 26 Jeff Erickson. Shortest Non-trivial Cycles in Directed Surface Graphs. In *Proc. 27th Ann. Symp. Comput. Geom.*, pages 236–243, 2011. doi:10.1145/1998196.1998231.
- 27 Jeff Erickson, Kyle Fox, and Luvsandondov Lkhamsuren. Holiest minimum-cost paths and flows in surface graphs. In *Proc. 50th Ann. ACM Symp. Theory Comput.*, pages 1319–1332, 2018. doi:10.1145/3188745.3188904.
- 28 Jeff Erickson and Sarel Har-Peled. Optimally cutting a surface into a disk. *Discrete Comput. Geom.*, 31(1):37–59, 2004. doi:10.1007/s00454-003-2948-z.
- 29 Jeff Erickson and Amir Nayyeri. Minimum cuts and shortest non-separating cycles via homology covers. In *Proc. 22nd Ann. ACM-SIAM Symp. Discrete Algorithms*, pages 1166–1176, 2011. doi:10.1137/1.9781611973082.88.
- 30 Jeff Erickson and Yipu Wang. Topologically Trivial Closed Walks in Directed Surface Graphs. Preprint, March 2019. arXiv:1812.01564.
- 31 Jeff Erickson and Kim Whittlesey. Transforming Curves on Surfaces redux. In *Proc. 24th Ann. ACM-SIAM Symp. Discrete Algorithms*, pages 1646–1655, 2013. doi:10.1137/1.9781611973105.118.
- 32 William J. Floyd and Steven J. Plotnick. Growth functions on Fuchsian groups and the Euler characteristic. *Invent. Math.*, 88(1):1–29, 1987. doi:10.1007/BF01405088.
- 33 Kyle Fox. Shortest Non-trivial Cycles in Directed and Undirected Surface Graphs. In *Proc. 24th Ann. ACM-SIAM Symp. Discrete Algorithms*, pages 352–364, 2013. doi:10.1137/1.9781611973105.26.
- 34 Steve M. Gersten and Hamish B. Short. Small cancellation theory and automatic groups. *Invent. Math.*, 102:305–334, 1990. doi:10.1007/BF01233430.
- 35 Peter Gublin. *Graphs, Surfaces and Homology*. Cambridge Univ. Press, 3rd edition, 2010.
- 36 Rostislav Grigorchuk and Pierre de la Harpe. On problems related to growth, entropy and spectrum in group theory. *J. Dynam. Control Systems*, 3(1):51–89, 1997. doi:10.1007/BF02471762.
- 37 Mikhail Gromov. Hyperbolic Groups. In Steve M. Gersten, editor, *Essays in Group Theory*, number 8 in Math. Sci. Res. Inst. Pub., pages 75–265. Springer-Verlag, 1987.
- 38 Joel Hass and Peter Scott. Intersections of curves on surfaces. *Israel J. Math.*, 51:90–120, 1985. doi:10.1007/BF02772960.
- 39 Allen Hatcher. *Algebraic Topology*. Cambridge Univ. Press, 2002. URL: <http://www.math.cornell.edu/~hatcher/AT/ATpage.html>.
- 40 Kazuo Iwano and Kenneth Steiglitz. A semiring on convex polygons and zero-sum cycle problems. *SIAM J. Comput.*, 19(5):883–901, 1990. doi:10.1137/0219061.
- 41 Ming-Yang Kao. Linear-processor NC algorithms for planar directed graphs I: Strongly connected components. *SIAM J. Comput.*, 22(3):431–459, 1993. doi:10.1137/0222032.
- 42 Ming-Yang Kao and Philip N. Klein. Toward overcoming the transitive-closure bottleneck: Efficient parallel algorithms for planar digraphs. *J. Comput. Syst. Sci.*, 47(3):459–500, 1993. doi:10.1016/0022-0000(93)90042-U.

- 43 Ming-Yang Kao and Gregory E. Shannon. Local reorientation, global order, and planar topology. In *Proc. 21st Ann. ACM Symp. Theory Comput.*, pages 286–296, 1989. doi:10.1145/73007.73034.
- 44 Ming-Yang Kao and Gregory E. Shannon. Linear-processor NC algorithms for planar directed graphs II: Directed spanning trees. *SIAM J. Comput.*, 22(3):460–481, 1993. doi:10.1137/0222033.
- 45 S. Rao Kosaraju and Gregory F. Sullivan. Detecting cycles in dynamic graphs in polynomial time (preliminary version). In *Proc. 20th ACM Symp. Theory. Comput.*, pages 398–406, 1988. doi:10.1145/62212.62251.
- 46 Martin Kutz. Computing shortest non-trivial cycles on orientable surfaces of bounded genus in almost linear time. In *Proc. 22nd Ann. Symp. Comput. Geom.*, pages 430–438, 2006. doi:10.1145/1137856.1137919.
- 47 Francis Lazarus and Julien Rivaud. On the homotopy test on surfaces. In *Proc. 53rd Ann. IEEE Symp. Foundations Comput. Sci.*, pages 440–449, 2012. doi:10.1109/FOCS.2012.12.
- 48 John Milnor. A note on curvature and the fundamental group. *J. Diff. Geom.*, 2(1):1–7, 1968. doi:10.4310/jdg/1214501132.
- 49 Bojan Mohar and Carsten Thomassen. *Graphs on Surfaces*. Johns Hopkins Univ. Press, 2001.
- 50 Shay Mozes, Kirill Nikolaev, Yahav Nussbaum, and Oren Weimann. Minimum Cut of Directed Planar Graphs in $O(n \log \log n)$ Time. In *Proc. 29th Ann. ACM-SIAM Symp. Discrete Algorithms*, pages 477–494, 2018. doi:10.1137/1.9781611975031.32.
- 51 David E. Muller and Paul E. Schupp. Groups, the theory of ends, and context-free languages. *J. Comput. System Sci.*, 26(3):295–310, 1983. doi:10.1016/0022-0000(83)90003-X.
- 52 August F. Möbius. Über der Bestimmung des Inhaltes eines Polyeders. *Ber. Sächs. Akad. Wiss. Leipzig, Math.-Phys. Kl.*, 17:31–68, 1865. *Gesammelte Werke* 2:473–512, Leipzig, 1886.
- 53 Vijaya Ramachandran and Honghua Yang. Finding the closed partition of a planar graph. *Algorithmica*, 11(5):443–468, 1994. doi:10.1007/BF01293266.
- 54 L. B. Smikun. Connection between context-free groups and groups with decidable problems of automata equivalence. *Cybernetics*, 12(5):687–691, 1976. Translated from *Kibernetika (Kiev)* (5):33–37, 1976. doi:10.1007/BF01070252.
- 55 Carsten Thomassen. Embeddings of graphs with no short noncontractible cycles. *J. Comb. Theory Ser. B*, 48(2):155–177, 1990. doi:10.1016/0095-8956(90)90115-G.
- 56 Mihalis Yannakakis. Graph-theoretic methods in database theory. In *Proc. 9th ACM SIGACT-SIGMOD-SIGART Symp. Principles Database Syst.*, pages 230–242, 1990. doi:10.1145/298514.298576.